3 speed Rotary Encoder Full Step library for Arduino

This is an optimized three speed Rotary Encoder library for Arduino which supports:

- Full step Rotary Encoder types.
- Detect three rotation speeds.
- Configurable sensitivity.
- Polling and interrupts.
- Optional button.

Hardware

Connect the two rotary pins to the DIGITAL pins of an Arduino board.

A third rotary button pin is not used in the Rotary library, but can be used in the sketch.

Tested with Arduino IDE v1.8.5 on hardware:

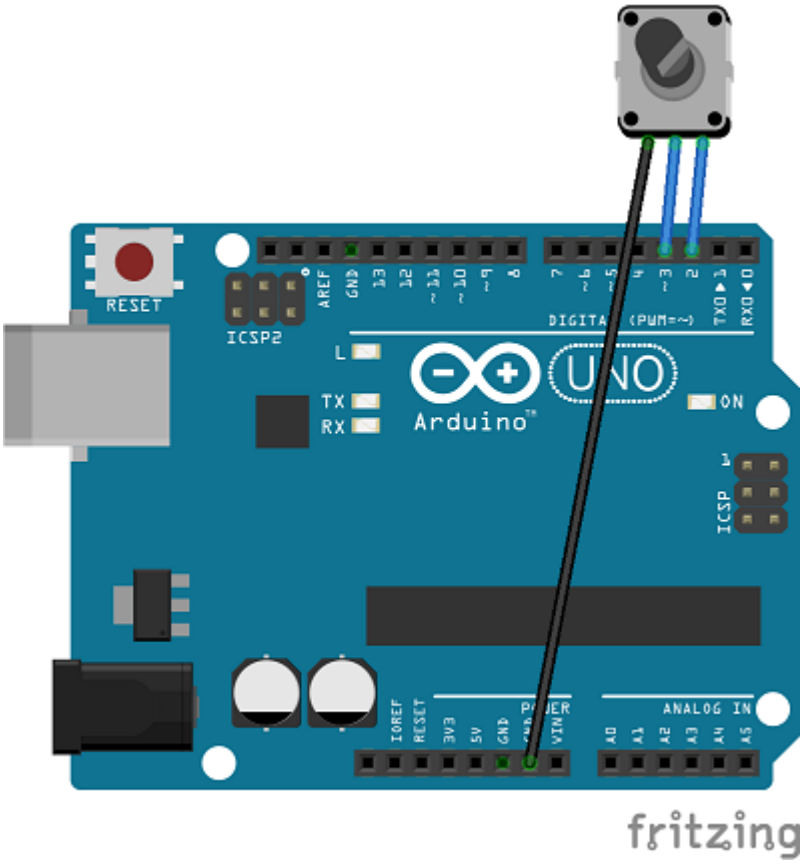
- Arduino UNO
- Arduino Nano
- Arduino Micro
- Arduino Pro or Pro Mini
- Arduino Mega or Mega2560
- Arduino Leonardo
- WeMos D1 R2 & mini

Interrupts

Both rotary pins must be connected to a DIGITAL pin with interrupt support, such as `INT0` or `INT1`. This is chip specific. Please refer to the documentation of your board or attachInterrupt().

Arduino UNO example

The connection below can be used for polled and interrupts. An optional button pin can be connected to DIGITAL pin 4.



Examples

The following examples are available:

- Rotary | Interrupt | InterruptFullStepBasic
- Rotary | Interrupt | InterruptFullStepButton
- Rotary | Interrupt | InterruptFullStepCounter
- Rotary | Polled | PolledFullStepBasic
- Rotary | Polled | PolledFullStepButton
- Rotary | Polled | PolledFullStepCounter

Usage

Initialization

```
#include <RotaryFullStep.h>

// Configure rotary pins connected to your Arduino board
#define ROTARY_PIN1   2
#define ROTARY_PIN2   3

// Initialize full step rotary encoder, default pull-up enabled, default
// sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2);

```

```
// Or initialize full step rotary encoder, pull-up disabled, default sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, false);

// Or initialize full step rotary encoder, pull-up enabled, sensitive 1..255
// A higher value is more sensitive
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, true, 150);
```

Read rotary with polling

```
void loop()
{
    int rotaryState = rotary.read();

    // rotaryState = -3: Turn left fastest
    // rotaryState = -2: Turn left faster
    // rotaryState = -1: Turn left
    // rotaryState = 0:  No change
    // rotaryState = 1:  Turn right
    // rotaryState = 2:  Turn right faster
    // rotaryState = 3:  Turn right fastest
}
```

Read rotary with interrupts

```
#include <Rotary.h>

// Connect rotary to Arduino DIGITAL pins with interrupt support:
//
// +----------------------------------+--------------------------+
// |              Board               |  DIGITAL interrupt pins  |
// +----------------------------------+--------------------------+
// | Uno, Nano, Mini, other 328-based |  2, 3                    |
// | Mega, Mega2560, MegaADK          |  2, 3, 18, 19, 20, 21    |
// | Micro, Leonardo, other 32u4-based |  0, 1, 2, 3, 7           |
// +----------------------------------+--------------------------+
//
#define ROTARY_PIN1   2
#define ROTARY_PIN2   3

// Initialize full step rotary encoder, default pull-up enabled, default
// sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2);

// Or initialize full step rotary encoder, pull-up disabled, default sensitive=100
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, false);

// Or initialize full step rotary encoder, pull-up enabled, sensitive 1..255
// A higher value is more sensitive
RotaryFullStep rotary(ROTARY_PIN1, ROTARY_PIN2, true, 150);

void setup()

{
```

```
  // Initialize pin change interrupt on both rotary encoder pins
  attachInterrupt(digitalPinToInterrupt(ROTARY_PIN1), rotaryInterrupt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(ROTARY_PIN2), rotaryInterrupt, CHANGE);
}

void rotaryInterrupt()
{
  int rotaryState = rotary.read();

  // rotaryState = -3: Turn left fastest
  // rotaryState = -2: Turn left faster
  // rotaryState = -1: Turn left
  // rotaryState = 0:  No change
  // rotaryState = 1:  Turn right
  // rotaryState = 2:  Turn right faster
  // rotaryState = 3:  Turn right fastest
}
```

Installation with Git

Install Git client for Windows

Install a Git client for Windows.

Install Git client for Linux

Open a command prompt and install a Git client for Linux, such as Debian Ubuntu:

```
sudo apt-get install git
```

Windows and Linux

The library must be installed in the Sketchbook directory which is configured in the Preferences dialog box.

1. Click `File` | `Preferences` | `Settings` tab and copy the Sketchbook location.
   The path on Windows is something like: `C:\Users\User\Documents\Arduino`
   The path on Linux is something like: `/home/user/Arduino`
2. Open a command prompt and type:

```
# Run on Windows:
cd C:\Users\User\Documents\Arduino
# Or run on Linux:
cd ~/Arduino

# Run the git clone library once:
git clone git clone https://github.com/Erriez/ErriezRotaryEncoderFullStep.git

# Update the library:
git pull
```

3. Restart the Arduino IDE.